EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

MODEL BUILDING
Save the Model

Date	02 November 2022
Team ID	PNT2022TMID26945
Project Name	Emerging Methods for Early Detection of Forest Fires.

##Importing The ImageDataGenerator Library

```
import keras
from keras.preprocessing.image import ImageDataGenerator
```

###Define the parameters/arguments for ImageDataGenerator class

```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoo
m_range=0.2,horizontal_flip=True)

test_datagen=ImageDataGenerator(rescale=1./255)
```

###Applying ImageDataGenerator Functionality to trainset

```
x_train=train_datagen.flow_from_directory(r'C:\archive\Dataset\Dataset\train_set',
target_size=(128,128),batch_size=32,class_mode='binary')
```

###Applying ImageDataGenerator Functionality to testset

```
x_test=test_datagen.flow_from_directory(r'C:\archive\Dataset\Dataset\test_set',tar
get_size=(128,128),batch_size=32,class_mode='binary')
```

```
##Import model building libraries

#To Define linear initialization import Sequential

from keras.models import Sequential

#To add layers import Dense

from keras.layers import Dense

#To create Convolution kernel import Convolution 2D

from keras.layers import Convolution2D

#import maxpooling layers

from keras.layers import MaxPooling2D

#import flatten Layer

from keras.layers import Flatten

import warnings

warnings.filterwarnings('ignore')

#Initializing the Model
model=Sequential()

##adding CNN layers
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

##adding maxpooling layer
model.add(MaxPooling2D(pool_size=(2,2)))

##adding flatten Layer
model.add(Flatten())
```

##add hidden layer

model.add(Dense(150,activation='relu'))

##add output layer

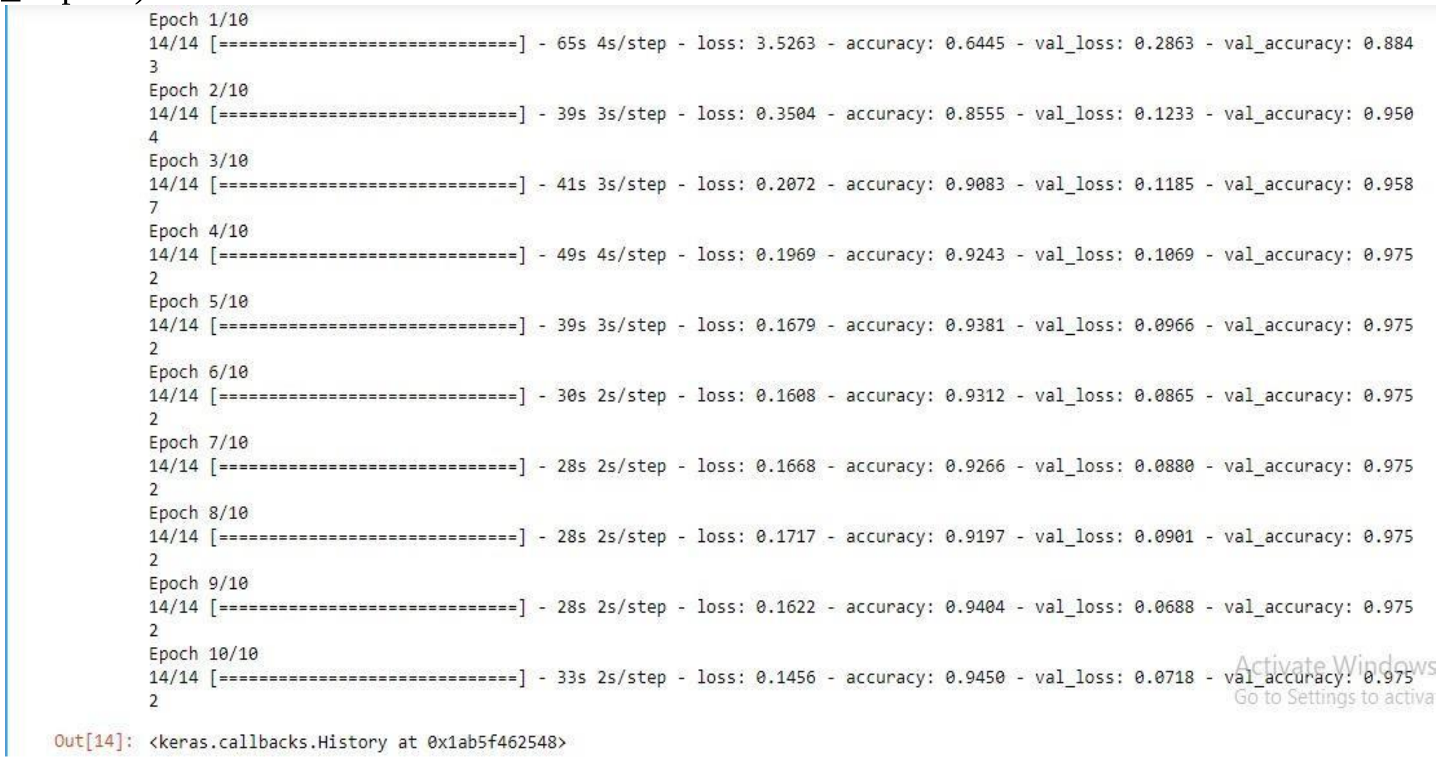
model.add(Dense(1,activation='sigmoid'))

#Configure the Learning Process

model.compile(loss="binary_crossentropy",optimizer="adam",metrics=['accuracy'])

## Training the model

model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)



```
          Epoch 1/10
          14/14 [==============================] - 65s 4s/step - loss: 3.5263 - accuracy: 0.6445 - val_loss: 0.2863 - val_accuracy: 0.884
          3
          Epoch 2/10
          14/14 [==============================] - 39s 3s/step - loss: 0.3504 - accuracy: 0.8555 - val_loss: 0.1233 - val_accuracy: 0.950
          4
          Epoch 3/10
          14/14 [==============================] - 41s 3s/step - loss: 0.2072 - accuracy: 0.9083 - val_loss: 0.1185 - val_accuracy: 0.958
          7
          Epoch 4/10
          14/14 [==============================] - 49s 4s/step - loss: 0.1969 - accuracy: 0.9243 - val_loss: 0.1069 - val_accuracy: 0.975
          2
          Epoch 5/10
          14/14 [==============================] - 39s 3s/step - loss: 0.1679 - accuracy: 0.9381 - val_loss: 0.0966 - val_accuracy: 0.975
          2
          Epoch 6/10
          14/14 [==============================] - 30s 2s/step - loss: 0.1608 - accuracy: 0.9312 - val_loss: 0.0865 - val_accuracy: 0.975
          2
          Epoch 7/10
          14/14 [==============================] - 28s 2s/step - loss: 0.1668 - accuracy: 0.9266 - val_loss: 0.0880 - val_accuracy: 0.975
          2
          Epoch 8/10
          14/14 [==============================] - 28s 2s/step - loss: 0.1717 - accuracy: 0.9197 - val_loss: 0.0901 - val_accuracy: 0.975
          2
          Epoch 9/10
          14/14 [==============================] - 28s 2s/step - loss: 0.1622 - accuracy: 0.9404 - val_loss: 0.0688 - val_accuracy: 0.975
          2
          Epoch 10/10
          14/14 [==============================] - 33s 2s/step - loss: 0.1456 - accuracy: 0.9450 - val_loss: 0.0718 - val_accuracy: 0.975
          2

   Out[14]: <keras.callbacks.History at 0x1ab5f462548>
```

#save the model

model.save("forest1.h5")